

Normas

- El examen puntúa sobre ?? puntos.
- La duración total del mismo es de **35 minutos**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.

Cuestionario

- (1 punto) 1. Indique la salida del siguiente programa (solamente el número de líneas, ninguna, una, dos ,tres, cuatro, etc, innumerables):

```
#include <stdio.h>
int main( int argc, char ** argv ) {
    int i ;
    for(i =4 ; i; i -=2){
        printf( "%d\n", i );
    }
    return( 0 );
}
```

Solución: Dos líneas

- (1 punto) 2. Sea el siguiente extracto de programa en lenguaje C:

```
#include <stdlib.h>
int main() {
    int **p;
    if ( ( p = malloc( sizeof (int **) ) ) == NULL ) {
        exit( 1 );
    }
    if ( ( *p = malloc( sizeof (int **) ) ) == NULL ) {
        exit( 1 );
    }
    /* ... */
    /* liberar memoria dinámica */
    /* POSIBLE LÍNEA QUE FALTA */
    free( p );
    return 0;
}
```

En la parte final se libera la memoria dinámica. Indique, si es el caso, una posible instrucción que puede faltar para liberar completamente la memoria dinámica asignada.

Solución: `free(*p);`

- (1 punto) 3. Sea el siguiente programa *prog_args.c* en lenguaje C:

```
#include <stdio.h>
int main( int argc, char **argv ) {

    printf( "%d\n", argc );
    printf( "%s\n", argv[ 0 ] );
    return 0;
}
```

Indique la salida que se obtiene al ejecutar el ejecutable asociado:

`./prog_args`

Solución: 1

`./prog_args`

- (1 punto) 4. Indique el comando del depurador 'gdb' que permite establecer un punto parada en un programa.

Solución:

break

- (1 punto) 5. Escriba las dos líneas que escribiría por pantalla el siguiente programa en C :

```
#include <stdio.h>
int main (int argc, char*argv[])
{

    int lista[]= {1,2,3,4,5};
    int *p;
    p=&lista[0];
    printf(" *p_=_%d\n", *(p+1));
    printf(" *P_=_%d\n", p[2]);

    return 0;
}
```

Solución:

`*p = 2`

`*P = 3`

- (1 punto) 6. Escribir una única sentencia que permitiría en C hacer una reserva de memoria dinámica para 300 enteros y todas estas posiciones se inicialicen con el valor 0.

Solución:

```
puntero = (int *) calloc(300, sizeof(int));
```

- (1 punto) 7. Escriba la salida que genera el siguiente programa en C :

```
#include <stdio.h>

int main (int argc, char*argv[])
{

    int x=11;

    while (x>0) {

        printf(" %d_",x);
        x=x/2;
    }

    return 0;

}
```

Solución: 11 5 2 1

- (1 punto) 8. Escriba la salida que genera el siguiente programa en C :

```
#include <stdio.h>

int main (int argc, char*argv[])
{

    int i=0, v=4;
    while (i<v) {

        switch(i)
        {

            case 1: printf("A\n");
            case 2: printf("B\n");break;
            case 3: printf("C"); break;
            default: printf("D");
        }

    }

}
```

```
        i++;  
    }  
  
    return 0;  
}
```

Solución: DA

B

B

C